The Improved Relevance Voxel Machine

Ganz, Melanie; Sabuncu, Mert; Van Leemput, Koen

Publication date:
2013

Document Version
Publisher's PDF, also known as Version of record

The Improved Relevance Voxel Machine

DTU Compute Technical Report-2013-10

Melanie Ganz[1,2], Mert Sabuncu[1], and Koen van Leemput[1,3,4]

[1] Martinos Center for Biomedical Imaging, Massachusetts General Hospital and Harvard Medical School, USA,
[2] Department for Computer Science, University of Copenhagen, Denmark
[3] Department of Applied Mathematics and Computer Science, Technical University of Denmark, Denmark
[4] Departments of Information and Computer Science and of Biomedical Engineering and Computational Science, Aalto University, Finland

1 Introduction

The concept of sparse Bayesian learning has received much attention in the machine learning literature as a means of achieving parsimonious representations of features used in regression and classification. It is an important family of algorithms for sparse signal recovery and compressed sensing and enables basis selection from overcomplete dictionaries.
One of the trailblazers of Bayesian learning is MacKay who already worked on the topic in his PhD thesis in 1992 [1]. Later on Tipping and Bishop developed the concept of sparse Bayesian learning [2, 3] and Tipping published the Relevance Vector Machine (RVM) [4] in 2001. While the concept of RVM was intriguing, problems with the approach were the run time which is approximately cubic in the number of basis functions as well as the greedy optimization. Hence, different approaches to overcome these shortcomings were developed, e.g. [5] or [6] as well as Tipping himself in [7] (FastRVM).
Recently, Sabuncu and Van Leemput [8, 9] extended the relevance vector machine by incorporating an additional spatial regularization term in the Gaussian prior on the regression weights or classification features (RVoxM). RVoxM encourages spatial clustering of the relevance voxels and computes predictions as linear combinations of their content. While the model of RVoxM produced nice results on age regression data [8, 9], the algorithm used a simple fixed point optimization scheme, which is not guaranteed to decrease the cost function at every step and is computationally expensive. In addition, RVoxM prunes voxels from the regression model by applying an artificial numerical threshold to the weight hyperparameters and hence has a free parameter that influences model sparsity. Finally, RVoxM can only remove voxels from the model, but not reintroduce them later on. Hence in its current form it is reminiscent of a greedy forward feature selection algorithm.
In this report, we aim to solve the problems of the original RVoxM algorithm in the spirit of [7] (FastRVM). We call the new algorithm Improved Relevance Voxel Machine (IRVoxM). Our contributions are an improvement of the greedy optimization algorithm

employed in RVoxM by exploiting the form of the marginal likelihood function and deriving an analytic expression for the optimal hyperparameter of each voxel, given the current hyperparameter of all other voxels. This enables us to maximize the marginal likelihood function in a principled and efficient manner. As a result IRVoxM optimizes the objective function better during training and the resulting models predict better on unseen cases. Finally, IRVoxM enables us to flexibly add and/or remove voxels during the optimization procedure.

2 Regression with the Relevance Voxel Machine - RVoxM

We base IRVoxM on the same theoretical model as RVoxM [8, 9]. In the regression problem, the target variable t, e.g. age or clinical test score, is assumed to be Gaussian distributed:

$$p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}), \beta^{-1}), \tag{1}$$

with variance β^{-1} and mean $y(\mathbf{x}, \mathbf{w}) = \sum_{i=1}^{M-1} x_i w_i + w_M = \mathbf{w}^{\mathrm{T}}\mathbf{x}$, where $\mathbf{x} \in \mathbb{R}^M$ is a vector that represents the input data, e.g. an image, plus a constant element of one ($x_M = 1$), and $\mathbf{w} \in \mathbb{R}^M$ are regression weights.
We further assume a Gaussian prior on $\mathbf{w}$ with hyperparamters $\boldsymbol{\alpha}$ and λ of the form

$$p(\mathbf{w}|\boldsymbol{\alpha}, \lambda) = \mathcal{N}(\mathbf{w}|0, \mathbf{P}^{-1}), \tag{2}$$

where $\mathbf{P} = \mathrm{diag}(\boldsymbol{\alpha}) + \lambda\mathbf{K}$. $\mathbf{K} = \mathbf{\Gamma}^{\mathrm{T}}\mathbf{\Gamma}$ is the graph Laplacian matrix which is a sparse, symmetric matrix and can be defined as the inner product of the incidence matrix $\mathbf{\Gamma}$. $\mathbf{\Gamma}$ is a sparse matrix of dimension $N_{\mathrm{Edg}} \times M$, where N_{Edg} denotes the number of edges in the graph spanned by $\mathbf{K}$. Each row of $\mathbf{\Gamma}$ has only two entries that denote the outgoing $(+1)$ and incoming (-1) nodes of an edge in the graph. In our case, edges connect physically neighboring locations, e.g. all voxels in the 6-neighborhood are connected to a central voxel in a volumetric image, but the neighborhood could also be modified. $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_M)^{\mathrm{T}}$ and λ are hyperparameters; the α_i are inverse covariances of the weight prior and hence control the sparsity of the weights. A large α_i means the weight w_i of the associated voxel is tending to zero, while a small α_i implies that the value w_i is largely determined by its neighbors. The parameter λ encourages spatial smoothness and the larger it is the smoother the resulting weight maps are. A graphical model describing the regression model can be found in [9] and is re-printed in figure 1 for illustrative purposes.

2.1 Training

With the above prior, the hyperparameters can be estimated by maximizing the following type-II likelihood given a collection of training target values $\mathbf{t} = (t_1, \ldots, t_N)^{\mathrm{T}}$ and a set of N training images $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^{\mathrm{T}}$:

$$\begin{aligned} p(\mathbf{t}|\mathbf{X}, \boldsymbol{\alpha}, \beta, \lambda) &= \int_{\mathbf{w}} p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) p(\mathbf{w}|\boldsymbol{\alpha}, \lambda) d\mathbf{w} \\ &= \int_{\mathbf{w}} \left(\prod_{n=1}^{N} p(t_n|\mathbf{x}_n, \mathbf{w}, \beta) \right) p(\mathbf{w}|\boldsymbol{\alpha}, \lambda) d\mathbf{w} = \mathcal{N}(\mathbf{t}|\mathbf{0}, \mathbf{C}), \end{aligned} \tag{3}$$

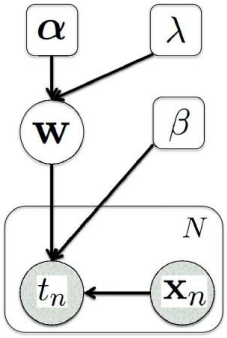

Fig. 1. A graphical model describing the RVoxM regression model taken from [9].

where we define $\mathbf{C} = \beta^{-1}\mathbf{I} + \mathbf{X}\mathbf{P}^{-1}\mathbf{X}^{\mathrm{T}}$. We can estimate the hyperparameters $\boldsymbol{\alpha}, \beta, \lambda$, which is equivalent to maximizing Eq. 4:

$$\hat{\boldsymbol{\alpha}}, \hat{\beta}, \hat{\lambda} = \underset{\boldsymbol{\alpha},\beta,\lambda}{\operatorname{argmax}}\, \mathcal{L}(\boldsymbol{\alpha}, \beta, \lambda) = \underset{\boldsymbol{\alpha},\beta,\lambda}{\operatorname{argmax}} \left(-\frac{1}{2}(N \ln(2\pi) + \ln|\mathbf{C}| + \mathbf{t}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{t}) \right). \tag{4}$$

Here, $\mathcal{L}(\boldsymbol{\alpha}, \beta, \lambda)$ denotes the logarithm of the marginal likelihood function, which is obtained by integrating out the weight parameters as shown in Eq. 3. In RVoxM [8], this optimization was solved by a coordinate ascent over β and λ, while optimizing over all $\boldsymbol{\alpha}$ simultaneously using a fixed point equation and a greedy approach, where single α_i's exceeding a numerical threshold are pruned from the model. This optimization of $\boldsymbol{\alpha}$ has no theoretical guarantees of convergence and is computationally expensive. Hence, we focus on deriving a better optimization algorithm for $\boldsymbol{\alpha}$.

2.2 Prediction

After obtaining $\hat{\boldsymbol{\alpha}}, \hat{\beta}, \hat{\lambda}$ from training data, we can make predictions for a new $\mathbf{x}^\star$ according to

$$p(t^\star|\mathbf{x}^\star, \mathbf{X}, \mathbf{t}, \hat{\boldsymbol{\alpha}}, \hat{\beta}, \hat{\lambda}) = \int p(t^\star|\mathbf{x}^\star, \mathbf{w}, \hat{\beta}) p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \hat{\boldsymbol{\alpha}}, \hat{\lambda}) d\mathbf{w} = \mathcal{N}(\mu^\star, \Sigma^\star), \tag{5}$$

where $p(t^\star|\mathbf{x}^\star, \mathbf{w}, \hat{\beta})$ is given by the regression model in Eq. 1 and $\mu^\star = \boldsymbol{\mu}^T\mathbf{x}$ and $\Sigma^\star = \frac{1}{\beta} + \mathbf{x}^{\mathrm{T}}\boldsymbol{\Sigma}\mathbf{x}$, in which $\boldsymbol{\Sigma} = (\mathbf{P} + \beta\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}$ and $\boldsymbol{\mu} = \boldsymbol{\Sigma}\mathbf{X}^{\mathrm{T}}\mathbf{t}$.

3 The Improved Relevance Voxel Machine - IRVoxM

To derive the improved Relevance Voxel Machine (IRVoxM) we start with the logarithm of the marginal likelihood function $\mathcal{L}(\boldsymbol{\alpha}, \beta, \lambda)$ for fixed β and λ; thus $\mathcal{L}(\boldsymbol{\alpha}, \beta, \lambda)$ is only

dependent on $\boldsymbol{\alpha}$. We use $\mathcal{L}(\alpha)$ from eq. 4 and rewrite it in the following way:

$$\begin{aligned}\mathcal{L}(\alpha) &= -\frac{1}{2}\left(N\ln(2\pi) + \ln|\mathbf{C}| + \mathbf{t}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{t}\right) \\ &= -\frac{1}{2}\left(N\ln(2\pi) + \ln(\beta^{-N}\frac{|\boldsymbol{\Sigma}|}{|\mathbf{P}|}) + \mathbf{t}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{t}\right) \end{aligned} \tag{6}$$

since $|\mathbf{C}| = |\beta^{-1}\mathbf{I} + \mathbf{X}\mathbf{P}^{-1}\mathbf{X}^{\mathrm{T}}| = \frac{|\beta^{-1}\mathbf{I}|\cdot|\beta\mathbf{X}\mathbf{X}^{\mathrm{T}}+\mathbf{P}|}{|\mathbf{P}|} = \frac{|\beta^{-1}\mathbf{I}|\cdot||\boldsymbol{\Sigma}|}{|\mathbf{P}|} = \beta^{-N}\frac{||\boldsymbol{\Sigma}|}{|\mathbf{P}|}$. Next we add additional terms that equal one and re-formulate the cost function further:

$$\begin{aligned}&\mathcal{L}(\alpha) - \frac{1}{2}\left(N\ln(2\pi) + \ln(\beta^{-N}|\boldsymbol{\Sigma}|\cdot\frac{|\mathrm{diag}(\boldsymbol{\alpha})|}{|\mathrm{diag}(\boldsymbol{\alpha})|}\cdot\frac{\lambda^{N_{\mathrm{Edg}}}}{\lambda^{N_{\mathrm{Edg}}}}) - \ln|\mathbf{P}| + \mathbf{t}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{t}\right) \\ &= -\frac{1}{2}\left(N\ln(2\pi) + \ln(\beta^{-N}\lambda^{-N_{\mathrm{Edg}}}\frac{|\boldsymbol{\Sigma}|}{|\mathrm{diag}(\boldsymbol{\alpha})|}) + N_{\mathrm{Edg}}\ln\lambda + \ln|\mathrm{diag}(\boldsymbol{\alpha})| - \ln|\mathbf{P}| + \mathbf{t}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{t}\right) \\ &= -\frac{1}{2}\left(N\ln(2\pi) + \ln|\tilde{\mathbf{C}}| + N_{\mathrm{Edg}}\ln\lambda + \ln|\mathrm{diag}(\boldsymbol{\alpha})| - \ln|\mathbf{P}| + \tilde{\mathbf{t}}^{\mathrm{T}}\tilde{\mathbf{C}}^{-1}\tilde{\mathbf{t}}\right) \\ &= -\frac{1}{2}\left(N\ln(2\pi) + \ln|\tilde{\mathbf{C}}| + \tilde{\mathbf{t}}^{\mathrm{T}}\tilde{\mathbf{C}}^{-1}\tilde{\mathbf{t}} + N_{\mathrm{Edg}}\ln\lambda + \ln|\mathrm{diag}(\boldsymbol{\alpha})| - \ln|\mathbf{P}|\right),\end{aligned}$$

where we have used the substitutions $|\tilde{\mathbf{C}}| = \beta^{-N}\lambda^{-N_{\mathrm{Edg}}}\frac{|\boldsymbol{\Sigma}|}{|\mathrm{diag}(\boldsymbol{\alpha})|}$ and $\tilde{\mathbf{C}}^{-1} = \tilde{\mathbf{B}} - \tilde{\mathbf{B}}\tilde{\mathbf{X}}\tilde{\boldsymbol{\Sigma}}\tilde{\mathbf{X}}^{\mathrm{T}}\tilde{\mathbf{B}}$ as well as $\mathbf{t}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{t} = \tilde{\mathbf{t}}^{\mathrm{T}}\tilde{\mathbf{C}}^{-1}\tilde{\mathbf{t}}$, in which

$$\begin{aligned}\mathbf{X} \rightarrow \tilde{\mathbf{X}} &= \begin{pmatrix}\mathbf{X} \\ \boldsymbol{\Gamma}\end{pmatrix} \\ \mathbf{t} \rightarrow \tilde{\mathbf{t}} &= \begin{pmatrix}\mathbf{t} \\ \mathbf{0}\end{pmatrix} \\ \beta \rightarrow \tilde{\mathbf{B}} &= \begin{pmatrix}\beta\mathbf{I}_N & \mathbf{0} \\ \mathbf{0} & \lambda\mathbf{I}_M\end{pmatrix}\end{aligned}$$

Note, that $\boldsymbol{\Sigma} = (\mathbf{P}+\beta\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1} = (\mathrm{diag}(\boldsymbol{\alpha})+\lambda\boldsymbol{\Gamma}^{\mathrm{T}}\boldsymbol{\Gamma}+\beta\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1} = \left(\mathrm{diag}(\boldsymbol{\alpha}) + \tilde{\mathbf{X}}^{\mathrm{T}}\tilde{\mathbf{B}}\tilde{\mathbf{X}}\right)^{-1} = \tilde{\boldsymbol{\Sigma}}$ as well as $\boldsymbol{\mu} = \boldsymbol{\Sigma}\mathbf{X}^{\mathrm{T}}\mathbf{t} = \tilde{\boldsymbol{\Sigma}}\tilde{\mathbf{X}}^{\mathrm{T}}\tilde{\mathbf{t}} = \tilde{\boldsymbol{\mu}}$.

4 Speeding up RVoxM

In the next step, we examine the different terms in the logarithm of the marginal likelihood $\mathcal{L}(\alpha)$. As derived above our new $\tilde{\mathbf{C}}^{-1} = \tilde{\mathbf{B}} - \tilde{\mathbf{B}}\tilde{\mathbf{X}}\tilde{\boldsymbol{\Sigma}}\tilde{\mathbf{X}}^{\mathrm{T}}\tilde{\mathbf{B}}$ and hence by the Woodbury identity

$$\tilde{\mathbf{C}} = \tilde{\mathbf{B}}^{-1} + \tilde{\mathbf{X}}\mathrm{diag}(\boldsymbol{\alpha}^{-1})\tilde{\mathbf{X}}^{\mathrm{T}} \tag{7}$$

As in [7] we derive the following three identitites:

$$\begin{aligned}\tilde{\mathbf{C}} &= \tilde{\mathbf{C}}_{-i} + \tilde{\mathbf{X}}_i\alpha_i^{-1}\tilde{\mathbf{X}}_i^{\mathrm{T}} \\ |\tilde{\mathbf{C}}| &= |\tilde{\mathbf{C}}_{-i}|\cdot|\mathbf{I}_M + \alpha_i^{-1}\tilde{\mathbf{X}}_i\tilde{\mathbf{C}}_{-i}^{-1}\tilde{\mathbf{X}}_i^{\mathrm{T}}| \\ \tilde{\mathbf{C}}^{-1} &= \tilde{\mathbf{C}}_{-i}^{-1} - \frac{\tilde{\mathbf{C}}_{-i}^{-1}\tilde{\mathbf{X}}_i\tilde{\mathbf{X}}_i^{\mathrm{T}}\tilde{\mathbf{C}}_{-i}^{-1}}{\alpha_i + \tilde{\mathbf{X}}_i^{\mathrm{T}}\tilde{\mathbf{C}}_{-i}^{-1}\tilde{\mathbf{X}}_i}\end{aligned}$$

in which $\tilde{\mathbf{X}}_i$ denotes the i-th column of $\tilde{\mathbf{X}}$. Furthermore, using the matrix determinant lemma we can re-write $|\mathbf{P}|$ to be of the form

$$\begin{aligned} |\mathbf{P}| &= \left|\mathrm{diag}(\boldsymbol{\alpha}) + \lambda \boldsymbol{\Gamma}^{\mathrm{T}}\boldsymbol{\Gamma}\right| \\ &= \left(\prod_i \alpha_i\right) \left|\mathbf{I} + \lambda \boldsymbol{\Gamma}\mathrm{diag}(\boldsymbol{\alpha}^{-1})\boldsymbol{\Gamma}^{\mathrm{T}}\right| \\ &= \left(\prod_i \alpha_i\right) \left|\mathbf{I} + \sum_i \frac{\lambda}{\alpha_i}\boldsymbol{\Gamma}_i\boldsymbol{\Gamma}_i^{\mathrm{T}}\right|, \end{aligned} \tag{8}$$

where $\boldsymbol{\Gamma}_i$ denotes the ith column in the matrix $\boldsymbol{\Gamma}$. In the same spirit we can rewrite $|\mathrm{diag}(\boldsymbol{\alpha})|$ to

$$|\mathrm{diag}(\boldsymbol{\alpha})| = \prod_i \alpha_i \tag{9}$$

With these definitions, we can re-structure the logarithm of the marginal likelihood $\mathcal{L}(\alpha)$ and divide the contribution made by the $\alpha_{m,m\neq i}$ from the contribution made by

α_i:

$$
\begin{aligned}
&\mathcal{L}(\alpha) \\
&= -\frac{1}{2}\left(N\ln(2\pi) + \ln|\tilde{\mathbf{C}}| + \tilde{\mathbf{t}}^{\mathrm{T}}\tilde{\mathbf{C}}^{-1}\tilde{\mathbf{t}}\right) - \frac{1}{2}\left(N_{\mathrm{Edg}}\ln(\lambda) - \ln|\mathbf{P}| + \ln|\mathrm{diag}(\boldsymbol{\alpha})|\right) \\
&= -\frac{1}{2}\left(N\ln(2\pi) + \ln\left(|\tilde{\mathbf{C}}_{-i}|\cdot|\mathbf{I}_M + \alpha_i^{-1}\tilde{\mathbf{X}}_i^{\mathrm{T}}\tilde{\mathbf{C}}_{-i}^{-1}\tilde{\mathbf{X}}_i|\right) + \tilde{\mathbf{t}}^{\mathrm{T}}\left(\tilde{\mathbf{C}}_{-i}^{-1} - \frac{\tilde{\mathbf{C}}_{-i}^{-1}\tilde{\mathbf{X}}_i\tilde{\mathbf{X}}_i^{\mathrm{T}}\tilde{\mathbf{C}}_{-i}^{-1}}{\alpha_i + \tilde{\mathbf{X}}_i^{\mathrm{T}}\tilde{\mathbf{C}}_{-i}^{-1}\tilde{\mathbf{X}}_i}\right)\tilde{\mathbf{t}}\right) \\
&\quad -\frac{1}{2}\left(N_{\mathrm{Edg}}\ln(\lambda) - \ln\left[\left(\prod_i \alpha_i\right)\left|\mathbf{I} + \sum_i \frac{\lambda}{\alpha_i}\boldsymbol{\Gamma}_i\boldsymbol{\Gamma}_i^{\mathrm{T}}\right|\right] + \ln\prod_i \alpha_i\right) \\
&= -\frac{1}{2}\left(N\ln(2\pi) + \ln|\tilde{\mathbf{C}}_{-i}| + \tilde{\mathbf{t}}^{\mathrm{T}}\tilde{\mathbf{C}}_{-i}^{-1}\tilde{\mathbf{t}} + N_{\mathrm{Edg}}\ln(\lambda) + \ln\prod_i \alpha_i\right) \\
&\quad -\frac{1}{2}\left(\ln\left(|\alpha_i^{-1}(\alpha_i + \tilde{\mathbf{X}}_i^{\mathrm{T}}\tilde{\mathbf{C}}_{-i}^{-1}\tilde{\mathbf{X}}_i)|\right) - \tilde{\mathbf{t}}^{\mathrm{T}}\frac{\tilde{\mathbf{C}}_{-i}^{-1}\tilde{\mathbf{X}}_i\tilde{\mathbf{X}}_i^{\mathrm{T}}\tilde{\mathbf{C}}_{-i}^{-1}}{\alpha_i + \tilde{\mathbf{X}}_i^{\mathrm{T}}\tilde{\mathbf{C}}_{-i}^{-1}\tilde{\mathbf{X}}_i}\tilde{\mathbf{t}}\right) \\
&\quad -\frac{1}{2}\left(-ln\prod_i \alpha_i - \ln\left|\overbrace{\mathbf{I} + \sum_{j\neq i}\frac{\lambda}{\alpha_j}\boldsymbol{\Gamma}_j\boldsymbol{\Gamma}_j^{\mathrm{T}}}^{\boldsymbol{\Psi}_{-\mathbf{i}}} + \frac{\lambda}{\alpha_i}\boldsymbol{\Gamma}_i\boldsymbol{\Gamma}_i^{\mathrm{T}}\right|\right) \\
&= -\frac{1}{2}\left(N\ln(2\pi) + \ln|\tilde{\mathbf{C}}_{-i}| + \tilde{\mathbf{t}}^{\mathrm{T}}\tilde{\mathbf{C}}_{-i}^{-1}\tilde{\mathbf{t}} + N_{\mathrm{Edg}}\ln(\lambda)\right) \\
&\quad -\frac{1}{2}\left(\ln\left(|\alpha_i^{-1}(\alpha_i\mathbf{I}_M + \tilde{\mathbf{X}}_i^{\mathrm{T}}\tilde{\mathbf{C}}_{-i}^{-1}\tilde{\mathbf{X}}_i)|\right) - \tilde{\mathbf{t}}^{\mathrm{T}}\frac{\tilde{\mathbf{C}}_{-i}^{-1}\tilde{\mathbf{X}}_i\tilde{\mathbf{X}}_i^{\mathrm{T}}\tilde{\mathbf{C}}_{-i}^{-1}}{\alpha_i + \tilde{\mathbf{X}}_i^{\mathrm{T}}\tilde{\mathbf{C}}_{-i}^{-1}\tilde{\mathbf{X}}_i}\tilde{\mathbf{t}}\right) \\
&\quad -\frac{1}{2}\left(-\ln\left(|\boldsymbol{\Psi}_{-\mathbf{i}}|\left(1 + \overbrace{\lambda\boldsymbol{\Gamma}_i{\boldsymbol{\Psi}_{-\mathbf{i}}}^{-1}\boldsymbol{\Gamma}_i^{\mathrm{T}}}^{a_i}}{\alpha_i}\right)\right)\right) \\
&= -\frac{1}{2}\left(N\ln(2\pi) + \ln|\tilde{\mathbf{C}}_{-i}| + \tilde{\mathbf{t}}^{\mathrm{T}}\tilde{\mathbf{C}}_{-i}^{-1}\tilde{\mathbf{t}} + N_{\mathrm{Edg}}\ln(\lambda)\right) \\
&\quad -\frac{1}{2}\left(-\ln\alpha_i + \ln(\alpha_i + \tilde{s}_i) - \frac{\tilde{q}^2}{\alpha_i + \tilde{s}_i}\right) \\
&\quad -\frac{1}{2}\left(-\ln(|\boldsymbol{\Psi}_{-\mathbf{i}}|) - \ln(1 + \frac{a_i}{\alpha_i})\right) \\
&= -\frac{1}{2}\left(N\ln(2\pi) + \ln|\tilde{\mathbf{C}}_{-i}| + \tilde{\mathbf{t}}^{\mathrm{T}}\tilde{\mathbf{C}}_{-i}^{-1}\tilde{\mathbf{t}} + N_{\mathrm{Edg}}\ln(\lambda) - \ln(|\boldsymbol{\Psi}_{-\mathbf{i}}|)\right) \\
&\quad -\frac{1}{2}\left(+\ln(\alpha_i + \tilde{s}_i) - \frac{\tilde{q}^2}{\alpha_i + \tilde{s}_i} - \ln(\alpha_i + a_i)\right) \\
&= \mathcal{L}(\alpha_{-i}) + l(\alpha_i) \qquad (10)
\end{aligned}
$$

where we have defined

$$
\mathcal{L}(\alpha_{-i}) = -\frac{1}{2}\left(N\ln(2\pi) + \ln|\tilde{\mathbf{C}}_{-i}| + \tilde{\mathbf{t}}^{\mathrm{T}}\tilde{\mathbf{C}}_{-i}^{-1}\tilde{\mathbf{t}} + N_{\mathrm{Edg}}\ln(\lambda) - \ln(|\boldsymbol{\Psi}_{-\mathbf{i}}|)\right) \qquad (11)
$$

and

$$l(\alpha_i) = \frac{1}{2}\left(-\ln(\alpha_i + \tilde{s}_i) + \frac{\tilde{q}_i^2}{\alpha_i + \tilde{s}_i} + \ln(\alpha_i + a_i)\right) \tag{12}$$

as well as

$$\tilde{s}_i = \tilde{\mathbf{X}}_i^{\mathrm{T}} \tilde{\mathbf{C}}_{-i}^{-1} \tilde{\mathbf{X}}_i, \quad \tilde{q}_i = \tilde{\mathbf{X}}_i^{\mathrm{T}} \tilde{\mathbf{C}}_{-i}^{-1} \tilde{\mathbf{t}}, \tag{13}$$

$$\mathbf{\Psi}_{-\mathbf{i}} = \mathbf{I} + \sum_{j \neq i} \frac{\lambda}{\alpha_j} \mathbf{\Gamma}_j \mathbf{\Gamma}_j^{\mathrm{T}}, \quad a_i = \frac{\lambda \mathbf{\Gamma}_i {\mathbf{\Psi}_{-\mathbf{i}}}^{-1} \mathbf{\Gamma}_i^{\mathrm{T}}}{\alpha_i}. \tag{14}$$

Now we are in the position to identify optimal solutions for each α_i separately by examining $l(\alpha_i)$. If we calculate the derivatives of $l(\alpha_i)$, we arrive at:

$$\frac{\partial}{\partial \alpha_i} l(\alpha_i) = \frac{1}{2}\left(-\frac{1}{\alpha_i + \tilde{s}_i} - \frac{\tilde{q}_i^2}{(\alpha_i + \tilde{s}_i)^2} + \frac{1}{\alpha_i + a_i}\right) \tag{15}$$

The solutions of this, when we set it to zero and restrict us to $\alpha_i \geq 0$, are:

- $\alpha_1 = \infty$.
- $\alpha_2 = \frac{a_i(\tilde{s}_i + \tilde{q}_i^2) - \tilde{s}_i^2}{\tilde{s}_i - a_i - \tilde{q}_i^2}$

To ensure that our solutions are maxima, we derive the second derivative:

$$\frac{\partial^2}{\partial \alpha_i^2} l(\alpha_i) = \frac{1}{2}\left(\frac{1}{(\alpha_i + \tilde{s}_i)^2} + \frac{2\tilde{q}_i^2}{(\alpha_i + \tilde{s}_i)^3} - \frac{1}{(\alpha_i + a_i)^2}\right) \tag{16}$$

The second derivative evaluated at α_1 is equal to zero, so α_1 is neither a maximum nor a minimum. Evaluating the second derivative at α_2 yields:

$$\frac{\partial^2}{\partial \alpha_i^2} l(\alpha_i)|_{\alpha_2} = \frac{1}{2}\frac{\mathcal{A}\mathcal{B}}{\mathcal{C}} \tag{17}$$

where

$$\begin{aligned}
\mathcal{A} &= (a_i - \tilde{s}_i)^2 + 2(\tilde{s}_i - a_i - \tilde{q}_i^2) \cdot (a_i - \tilde{s}_i) - \tilde{q}_i^4 \\
\mathcal{B} &= (\tilde{s}_i - a_i - \tilde{q}_i^2)^2 \\
\mathcal{C} &= \tilde{q}_i^4 (\alpha_i - \tilde{s}_i)^4
\end{aligned}$$

$\mathcal{B}$ and $\mathcal{C}$ are always positive and $\mathcal{A}$ can be rewritten in the following way

$$\begin{aligned}
\mathcal{A} &= (a_i - \tilde{s}_i)^2 + 2(\tilde{s}_i - a_i - \tilde{q}_i^2) \cdot (a_i - \tilde{s}_i) - \tilde{q}_i^4 \\
&= -((a_i - \tilde{s}) + \tilde{q}_i^2)^2.
\end{aligned} \tag{18}$$

Thus, because $\mathcal{A} < 0$, $\mathcal{B} > 0$ and $\mathcal{C} > 0$, $\frac{\partial^2}{\partial \alpha_i^2} l(\alpha_i)|_{\alpha_2} = \frac{1}{2}\frac{\mathcal{A}\mathcal{B}}{\mathcal{C}}$ is always negative and α_2 is always a maximum.

5 How does the function look like?

To figure out how the function looks like, let us examine the limits of it at zero and infinity and the sign of the first derivative at infinity.

5.1 Case 1 - $a_i >= \tilde{s}_i$

If $a_i > \tilde{s}_i$ then $l(\alpha_i)$ is defined between $-\tilde{s}_i$ and $+\infty$. This has a pole at $-\tilde{s}_i$. The limits at the poles, 0 and $+\infty$, are

$$\begin{aligned}
\lim_{\alpha_i \to -\tilde{s}_i} l(\alpha_i) &= \frac{1}{2}\left(\ln(-\tilde{s}_i + a_i) - \ln(0) + \frac{\tilde{q}_i^2}{0}\right) \\
&= +\infty \\
\lim_{\alpha_i \to 0} l(\alpha_i) &= \frac{1}{2}\left(\ln(a_i) - \ln(\tilde{s}_i) + \frac{\tilde{q}_i^2}{\tilde{s}_i}\right) \\
&= const. \\
\lim_{\alpha_i \to \infty} l(\alpha_i) &= \lim_{\alpha_i \to \infty} \frac{1}{2}\left(\ln\left(\frac{\alpha_i + a_i}{\alpha_i + \tilde{s}_i}\right) + \frac{\tilde{q}_i^2}{\alpha_i + \tilde{s}_i}\right) \\
&= \frac{1}{2}\left(\lim_{\alpha_i \to \infty} \ln \frac{\alpha_i + a_i}{\alpha_i + \tilde{s}_i} + \lim_{\alpha_i \to \infty} \frac{\tilde{q}_i^2}{\alpha_i + \tilde{s}_i}\right) \\
&= \frac{1}{2}\left(\ln(\lim_{\alpha_i \to \infty}(1 + \frac{a_i - \tilde{s}_i}{\alpha_i + \tilde{s}_i})) + \lim_{\alpha_i \to \infty} \frac{\tilde{q}_i^2}{\alpha_i + \tilde{s}_i}\right) \\
&= 0
\end{aligned}$$

where $\lim_{\alpha_i \to 0} l(\alpha_i)$, the y-intercept of the function, is always positive.
The first derivative of $l(\alpha_i)$ can be re-written in the following way:

$$\begin{aligned}
\frac{\partial}{\partial \alpha_i} l(\alpha_i) &= \frac{1}{2}\left(-\frac{1}{\alpha_i + \tilde{s}_i} - \frac{\tilde{q}_i^2}{(\alpha_i + \tilde{s}_i)^2} + \frac{1}{\alpha_i + a_i}\right) \\
&= \frac{1}{2}\left(\frac{\alpha_i(\tilde{s}_i - a_i - \tilde{q}_i^2) - a_i(\tilde{s}_i - \tilde{q}_i^2) + \tilde{s}_i^2}{(\alpha_i + \tilde{s}_i)^2(\alpha_i + a_i)}\right) \\
&= \frac{1}{2}\left(\frac{(\tilde{s}_i - a_i - \tilde{q}_i^2) + \alpha_i^{-1})(\tilde{s}_i^2 - a_i(\tilde{s}_i - \tilde{q}_i^2))}{\alpha^{-1}(\alpha_i + \tilde{s}_i)^2(\alpha_i + a_i)}\right)
\end{aligned}$$

If we now examine the sign of the above at $\alpha = \infty$, we can see that the sign of the denominator is positive, while the numerator's sign is dependent on the term $\tilde{s}_i - a_i - \tilde{q}_i^2$. In our current case where $\tilde{s}_i - a_i < 0$ we get that $\text{sgn}(\tilde{s}_i - a_i - \tilde{q}_i^2) = -1$ and hence the curvature of $l(\alpha_i)$ at $\alpha = \infty$ is negative. If we evaluate the curvature at $\alpha = 0$ we get $\frac{\partial}{\partial \alpha_i} l(\alpha_i)|_0 = \frac{\tilde{s}_i(\tilde{s}_i - a_i) - \tilde{q}_i^2 a_i}{\tilde{s}_i^2 a_i}$ which is also always negative.

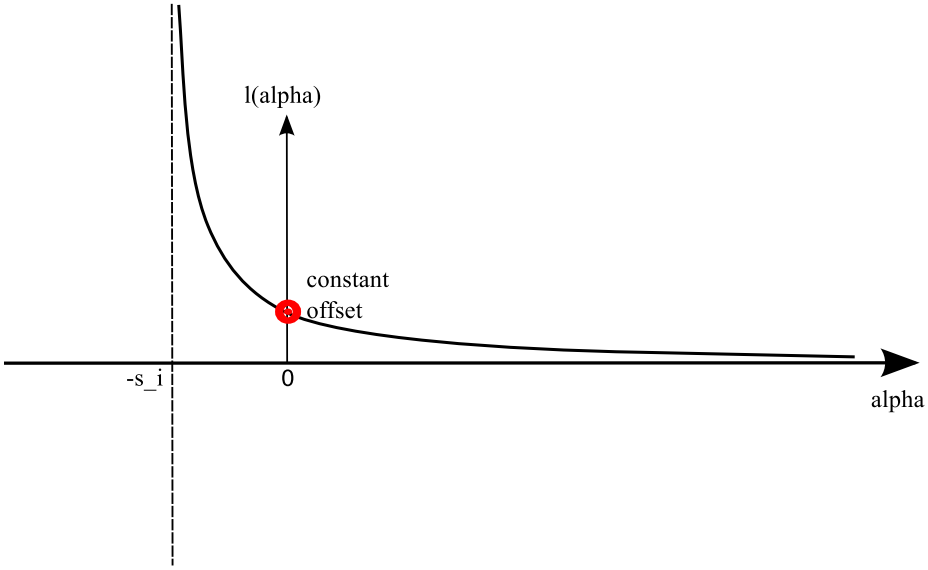

Fig. 2. $l(\alpha_i)$ for Case 1 - $a_i > \tilde{s}_i$

Furthermore, we can show that α_2 is always less than $-\tilde{s}_i$:

$$\begin{aligned}
\alpha_2 &= \frac{a_i(\tilde{s}_i + \tilde{q}_i^2) - \tilde{s}_i^2}{\tilde{s}_i - a_i - \tilde{q}_i^2} \\
&= \frac{-\tilde{s}_i(\tilde{s}_i - a_i) + a_i\tilde{q}_i^2}{\tilde{s}_i - a_i - \tilde{q}_i^2} \\
&= \frac{-\tilde{s}_i(\tilde{s}_i - a_i - \tilde{q}_i^2 + \tilde{q}_i^2) + a_i\tilde{q}_i^2}{\tilde{s}_i - a_i - \tilde{q}_i^2} \\
&= \frac{-\tilde{s}_i(\tilde{s}_i - a_i - \tilde{q}_i^2) + a_i\tilde{q}_i^2 - \tilde{s}_i\tilde{q}_i^2}{\tilde{s}_i - a_i - \tilde{q}_i^2} \\
&= -\tilde{s}_i + \frac{(a_i - \tilde{s}_i)\tilde{q}_i^2}{\tilde{s}_i - a_i - \tilde{q}_i^2}
\end{aligned}$$

Since the condition $a_i > \tilde{s}_i$ holds, the numerator of the second term is always positive, while the denominator is always negative and hence $\alpha_2 < -\tilde{s}_i$. This means our function looks like figure 2. In the positive range we can get the case where $l(0)$ is greater than $l(\infty)$.

5.2 Case 2 - $a_i < \tilde{s}_i$

If $a_i < \tilde{s}_i$, then $l(\alpha_i)$ is defined between $-a_i$ and $+\infty$. This has a pole at $-a_i$. The limits at the pole, 0 and $+\infty$ are

$$\begin{aligned}
\lim_{\alpha_i \to -a_i} l(\alpha_i) &= \frac{1}{2}\left(\ln(0) - \ln(-a_i + \tilde{s}_i) + \frac{\tilde{q}_i^2}{-a_i + \tilde{s}_i}\right) \\
&= -\infty \\
\lim_{\alpha_i \to 0} l(\alpha_i) &= \frac{1}{2}\left(\ln(a_i) - \ln(\tilde{s}_i) + \frac{\tilde{q}_i^2}{\tilde{s}_i}\right) \\
&= const. \\
\lim_{\alpha_i \to \infty} l(\alpha_i) &= \lim_{\alpha_i \to \infty} \frac{1}{2}\left(\ln\left(\frac{\alpha_i + a_i}{\alpha_i + \tilde{s}_i}\right) + \frac{\tilde{q}_i^2}{\alpha_i + \tilde{s}_i}\right) \\
&= \frac{1}{2}\left(\lim_{\alpha_i \to \infty} \ln \frac{\alpha_i + a_i}{\alpha_i + \tilde{s}_i} + \lim_{\alpha_i \to \infty} \frac{\tilde{q}_i^2}{\alpha_i + \tilde{s}_i}\right) \\
&= \frac{1}{2}\left(\ln(\lim_{\alpha_i \to \infty}(1 + \frac{a_i - \tilde{s}_i}{\alpha_i + \tilde{s}_i})) + \lim_{\alpha_i \to \infty} \frac{\tilde{q}_i^2}{\alpha_i + \tilde{s}_i}\right) \\
&= 0
\end{aligned}$$

If we now examine the sign of the first derivative of $l(\alpha_i)$ at $\alpha = \infty$ like above we have to make a distinction again:

Case 2.A - $\tilde{s}_i - a_i < \tilde{q}_i^2$ If $\tilde{s}_i - a_i < \tilde{q}_i^2$, then $\text{sgn}(\tilde{s}_i - a_i) - \tilde{q}_i^2 = -1$ and hence we have a single maximum at α_2 and then the function decreases to zero towards infinity. But the curvature at 0 can though be positive or negative. Looking at the sign or magnitude of α_2 for the case A does not yield any insight. α_2 can either be positive or negative.
This means our function looks like figure 3. In case 2A and for negative α we can again end up with $l(0)$ being greater than $l(\infty)$.

Case 2.B - $\tilde{s}_i - a_i >= \tilde{q}_i^2$ If $\tilde{s}_i - a_i > \tilde{q}_i^2$ then $\text{sgn}((\tilde{s}_i - a_i) - \tilde{q}_i^2) = +1$ and hence the curvature at $\alpha = \infty$ is positive. In addition, the curvature at $\alpha = 0$ is also positive, since $\frac{\partial}{\partial \alpha_i} l(\alpha_i)|_0 = \frac{\tilde{s}_i(\tilde{s}_i - a_i) - \tilde{q}_i^2 a_i}{\tilde{s}_i^2 a_i}$. The numerator determines the sign and can be lower bounded by $\tilde{s}_i(\tilde{s}_i - a_i) - (\tilde{s}_i - a_i)a_i = (\tilde{s}_i - a_i)^2 > 0$. Hence the curvature at $\alpha = 0$ is always positive.

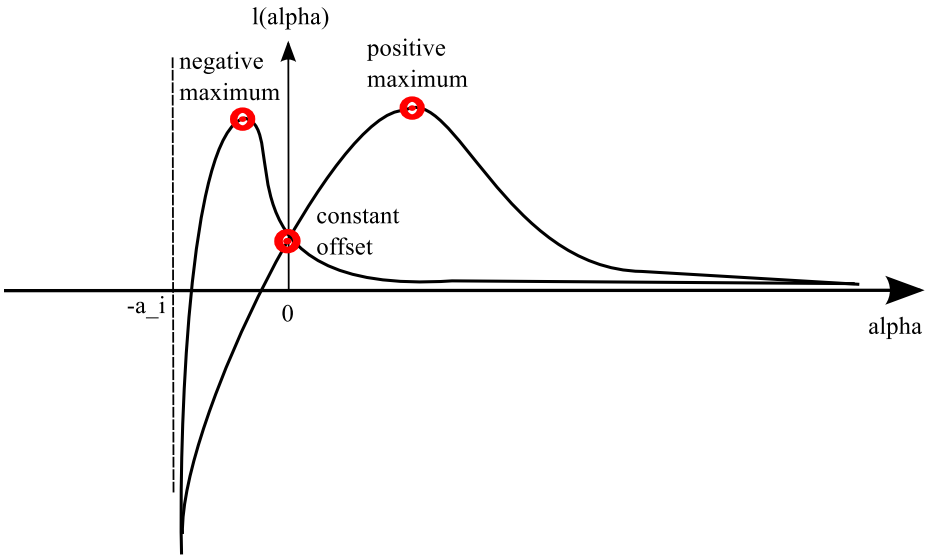

Fig. 3. $l(\alpha_i)$ for Case 2A - $a_i < \tilde{s}_i$ as well as $\tilde{s}_i - a_i < \tilde{q}_i^2$

Furthermore, we can show that α_2 is always smaller than $-a_i$:

$$\begin{aligned}
\alpha_2 &= \frac{a_i(\tilde{s}_i + \tilde{q}_i^2) - \tilde{s}_i^2}{\tilde{s}_i - a_i - \tilde{q}_i^2} \\
&= \frac{a_i(-\tilde{s}_i + 2\tilde{s}_i + a_i - a_i + \tilde{q}_i^2) - \tilde{s}_i^2}{\tilde{s}_i - a_i - \tilde{q}_i^2} \\
&= \frac{-a_i(\tilde{s}_i - a_i - \tilde{q}_i^2) + 2a_i\tilde{s}_i - a_i^2 - \tilde{s}_i^2}{\tilde{s}_i - a_i - \tilde{q}_i^2} \\
&= -a_i - \frac{(a_i - \tilde{s}_i)^2}{\tilde{s}_i - a_i - \tilde{q}_i^2}
\end{aligned}$$

Looking at the second term the numerator is always positive due to the square, while the denominator is always positive since $\tilde{s}_i - a_i > \tilde{q}_i^2$. Therefore, α_2 is always smaller than a_i. In addition, $\lim_{\alpha_i \to 0} l(\alpha_i) < 0$, since $\ln(a_i) - \ln(\tilde{s}_i) + \frac{\tilde{q}_i^2}{\tilde{s}_i}$ can be upper bounded by $\ln(\frac{a_i}{\tilde{s}_i}) + \frac{\tilde{s}_i - a_i}{\tilde{s}_i} \leq \frac{a_i}{\tilde{s}_i} - 1 + \frac{\tilde{s}_i - a_i}{\tilde{s}_i} = 0$.
Hence our likelihood looks like figure 4. There's a single maximum at $\alpha = \infty$.

5.3 Solution overview

In the following, we give a short overview over the different solutions that maximize the marginal likelihood.

- If $a_i >= \tilde{s}_i$, the solution that maximizes the marginal likelihood is $\alpha_i = 0$, since we have chosen $\alpha_i \geq 0$.

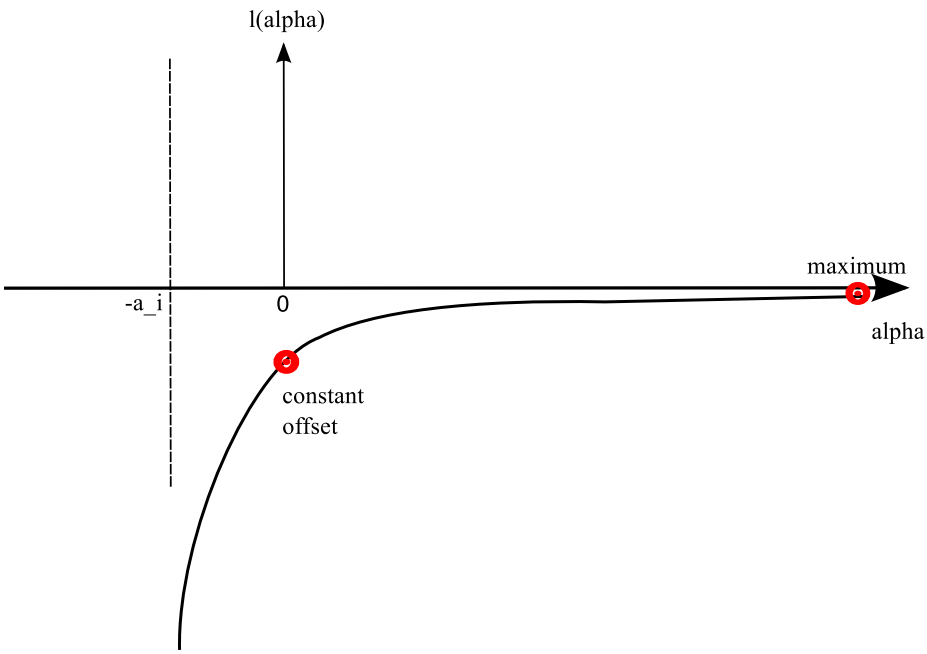

Fig. 4. $l(\alpha_i)$ for Case 2B - $a_i < \tilde{s}_i$ as well as $\tilde{s}_i - a_i > \tilde{q}_i^2$

- If $a_i < \tilde{s}_i$ and $\tilde{s}_i - a_i < \tilde{q}_i^2$, the solution that maximizes the marginal likelihood is given by $\alpha_i = \frac{a_i(\tilde{s}_i + \tilde{q}_i^2) - \tilde{s}_i^2}{\tilde{s}_i - a_i - \tilde{q}_i^2}$. If this solution is negative, the correct solution becomes $\alpha_i = 0$.
- If $a_i < \tilde{s}_i$ and $\tilde{s}_i - a_i >= \tilde{q}_i^2$, the solution that maximizes the marginal likelihood is to remove α_i from the model meaning $\alpha_i = \infty$.

6 Algorithm

6.1 No speedup

The original algorithm without any shortcuts or computational speedups is of the following form:

1. Initialize λ and β as well as initialize a starting model with all $\boldsymbol{\alpha}$ set to a value of 1 as was done in [9].
2. Randomly pick another voxel i.
3. Compute $\tilde{s}_i$,$\tilde{q}_i$ and a_i:

$$\tilde{s}_i = \tilde{\mathbf{X}}_i^{\mathrm{T}} \tilde{\mathbf{C}}_{-i}^{-1} \tilde{\mathbf{X}}_i \text{ and } \tilde{q}_i = \tilde{\mathbf{X}}_i^{\mathrm{T}} \tilde{\mathbf{C}}_{-i}^{-1} \tilde{\mathbf{t}} \tag{19}$$

where $\tilde{\mathbf{C}}_{-i}^{-1} = \left(\tilde{\mathbf{C}} - \tilde{\mathbf{X}}_i \alpha_i^{-1} \tilde{\mathbf{X}}^{\mathrm{T}} \right)^{-1}$ with $\tilde{\mathbf{C}} = \tilde{\mathbf{B}}^{-1} + \tilde{\mathbf{X}} \mathrm{diag}(\boldsymbol{\alpha}^{-1}) \tilde{\mathbf{X}}^{\mathrm{T}}$ as well as

$$a_i = \lambda \boldsymbol{\Gamma}_i \boldsymbol{\Psi}_{-\mathbf{i}}{}^{-1} \boldsymbol{\Gamma}_i^{\mathrm{T}} \tag{20}$$

where $\mathbf{\Psi}_{-\mathbf{i}} = \mathbf{I} + \sum_{j \neq i} \frac{\lambda}{\alpha_j} \mathbf{\Gamma}_j \mathbf{\Gamma}_j^{\mathrm{T}} = \mathbf{I} + \lambda \mathrm{diag}(\boldsymbol{\alpha}_{-i}^{-1}) \boldsymbol{\Gamma}_{-i} \boldsymbol{\Gamma}_{-i}^{\mathrm{T}}$.

4. Case 1 - $a_i \geq \tilde{s}_i$ leads to $\alpha_i = 0$.
5. Case 2 - $a_i < \tilde{s}_i$ leads to
   (a) Case A - $\tilde{s}_i - a_i < \tilde{q}_i^2$ leads to $\alpha_i = \frac{a_i(\tilde{s}_i + \tilde{q}_i^2) - \tilde{s}_i^2}{\tilde{s}_i - a_i - \tilde{q}_i^2}$. If the aforementioned α_i is negative the solution is $\alpha_i = 0$.
   (b) Case B - $\tilde{s}_i - a_i \geq \tilde{q}_i^2$ leads to $\alpha_i = \infty$.
6. After one has visited all voxels once in random order, we update β and λ by using the Matlab function fminbnd and out cost function to find the optimal value of β and λ.
7. Repeat from randomly picking a voxel until convergence is achieved.

6.2 Speedup 1

Since α_i can become zero, we need to re-formulate some of calculations to not involve $(\alpha_i)^{-1}$ in order to avoid numerical errors. Hence we substitute $\mathbf{\Psi}_{-\mathbf{i}} = \mathbf{I} + \lambda \mathrm{diag}(\boldsymbol{\alpha}_{-i}^{-1}) \boldsymbol{\Gamma}_{-i} \boldsymbol{\Gamma}_{-i}^{\mathrm{T}}$ with its Woodbury equivalent ${\mathbf{\Psi}_{-\mathbf{i}}}^{-1} = \mathbf{I} - \boldsymbol{\Gamma}_{-i} (\frac{1}{\lambda} \mathrm{diag}(\boldsymbol{\alpha}_{-i}) + \boldsymbol{\Gamma}_{-i}^{\mathrm{T}} \boldsymbol{\Gamma}_{-i})^{-1} \boldsymbol{\Gamma}_{-i}^{\mathrm{T}}$. Furthermore, since $\frac{1}{\lambda} \mathrm{diag}(\boldsymbol{\alpha}_{-i}) + \boldsymbol{\Gamma}_{-i}^{\mathrm{T}} \boldsymbol{\Gamma}_{-i}$ is very sparse, storing, updating, and inverting it directly is the fastest way to compute it.

6.3 Speedup 2

The next thing we will do is to speed up the calculation of $\tilde{s}_i$ and $\tilde{q}_i$. This can be done by using the same relation as given in eq. (23) of [7]:

$$\tilde{s}_m = \frac{\alpha_m \tilde{S}_m}{\alpha_m - \tilde{S}_m} \text{ and } \tilde{q}_m = \frac{\alpha_m \tilde{Q}_m}{\alpha_m - \tilde{S}_m} \tag{21}$$

where

$$\tilde{S}_m = \tilde{\mathbf{X}}_m^{\mathrm{T}} \tilde{\mathbf{C}}^{-1} \tilde{\mathbf{X}}_m \text{ and } \tilde{Q}_m = \tilde{\mathbf{X}}_m^{\mathrm{T}} \tilde{\mathbf{C}}^{-1} \tilde{\mathbf{t}} \tag{22}$$

For $\alpha_m = \infty$ we have $\tilde{s}_m = \tilde{S}_m$ as well as $\tilde{q}_m = \tilde{Q}_m$ whereas for $\alpha_m = 0$ we have $\tilde{s}_m = 0$ and $\tilde{q}_m = 0$. This way we do not need to keep track of different $\tilde{\mathbf{C}}_{-i}^{-1}$ and instead everything is based on $\tilde{\mathbf{C}}^{-1}$.

6.4 Speedup 3 - How we never need to compute $\tilde{\mathbf{C}}^{-1}$

Instead of making everything dependend on the direct computation of $\tilde{\mathbf{C}}^{-1}$, we can get around ever having to compute $\tilde{\mathbf{C}}^{-1}$ by following what was done in the appendix of [7] for all quantities. In addition we can then remove the $\tilde{}$ notation and re-write everything as much as we can in terms of the original variables. One calculates all the necessary quantities once in the beginnning and then changes them as described below when adding, changing or removing a basis function fom the model. Then the updates look like the following:

– First we initialize all variables:
$\tilde{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma} = (\mathrm{diag}(\boldsymbol{\alpha}) + \beta \mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda \boldsymbol{\Gamma}^{\mathrm{T}}\boldsymbol{\Gamma})^{-1}$

$$\tilde{\mathbf{C}}^{-1} = \begin{bmatrix} \tilde{\mathbf{C}}^{-1}_{(1,1)} & \tilde{\mathbf{C}}^{-1}_{(1,2)} \\ \tilde{\mathbf{C}}^{-1}_{(2,1)} & \tilde{\mathbf{C}}^{-1}_{(2,2)} \end{bmatrix} \tag{23}$$

where

$$\begin{aligned} \tilde{\mathbf{C}}^{-1}_{(1,1)} &= \mathrm{diag}(\boldsymbol{\beta}\mathbf{I}) - \beta^2 \mathbf{X}\boldsymbol{\Sigma}\mathbf{X}^{\mathrm{T}} \\ \tilde{\mathbf{C}}^{-1}_{(1,2)} &= -\beta \mathbf{X}\boldsymbol{\Sigma}\boldsymbol{\Gamma}^{\mathrm{T}}\lambda \\ \tilde{\mathbf{C}}^{-1}_{(2,1)} &= -\lambda \boldsymbol{\Gamma}\boldsymbol{\Sigma}\mathbf{X}^{\mathrm{T}}\beta \\ \tilde{\mathbf{C}}^{-1}_{(2,2)} &= \mathrm{diag}(\boldsymbol{\lambda}\mathbf{I}) - \lambda^2 \boldsymbol{\Gamma}\boldsymbol{\Sigma}\boldsymbol{\Gamma}^{\mathrm{T}} \end{aligned}$$

as well as $\tilde{\boldsymbol{\mu}} = \boldsymbol{\mu} = \boldsymbol{\Sigma}\mathbf{X}^{\mathrm{T}}\beta\mathbf{t}$ and

$$\begin{aligned} \tilde{\mathbf{S}}_m &= \mathbf{x}_m^{\mathrm{T}}\tilde{\mathbf{C}}^{-1}_{(1,1)}\mathbf{x}_m + \mathbf{x}_m^{\mathrm{T}}\tilde{\mathbf{C}}^{-1}_{(1,2)}\boldsymbol{\gamma}_m + \boldsymbol{\gamma}_m^{\mathrm{T}}\tilde{\mathbf{C}}^{-1}_{(2,1)}\mathbf{x}_m + \boldsymbol{\gamma}_m^{\mathrm{T}}\tilde{\mathbf{C}}^{-1}_{(2,2)}\boldsymbol{\gamma}_m \\ \tilde{\mathbf{Q}}_m &= \mathbf{x}_m^{\mathrm{T}}\tilde{\mathbf{C}}^{-1}_{(1,1)}\mathbf{t} + \mathbf{x}_m^{\mathrm{T}}\tilde{\mathbf{C}}^{-1}_{(1,2)}\mathbf{t} \end{aligned}$$

– Update the quantities when adding a new basis function like this:

$$\boldsymbol{\Sigma} \rightarrow \boldsymbol{\Sigma}_{+i} = \begin{bmatrix} \boldsymbol{\Sigma}_{+i}(1,1) & \boldsymbol{\Sigma}_{+i}(1,2) \\ \boldsymbol{\Sigma}_{+i}(2,1) & \boldsymbol{\Sigma}_{+i}(2,2) \end{bmatrix} \tag{24}$$

where

$$\begin{aligned} \boldsymbol{\tau} &= \beta \mathbf{x}_i^{\mathrm{T}}\mathbf{X} + \lambda \boldsymbol{\gamma}_i^{\mathrm{T}}\boldsymbol{\Gamma} \\ \boldsymbol{\Sigma}_{+i}(1,1) &= \boldsymbol{\Sigma} + \boldsymbol{\Sigma}\boldsymbol{\tau}^{\mathrm{T}}\tilde{\Sigma}_{ii}\boldsymbol{\tau}\boldsymbol{\Sigma} \\ \boldsymbol{\Sigma}_{+i}(2,1) &= -\tilde{\Sigma}_{ii}\boldsymbol{\tau}\boldsymbol{\Sigma} \\ \boldsymbol{\Sigma}_{+i}(1,2) &= -\boldsymbol{\Sigma}\boldsymbol{\tau}^{\mathrm{T}}\tilde{\Sigma}_{ii} \\ \boldsymbol{\Sigma}_{+i}(2,2) &= \tilde{\Sigma}_{ii} \end{aligned}$$

as well as $\tilde{\Sigma}_{ii} = (\alpha_i + \mathbf{x}_i^{\mathrm{T}}\tilde{\mathbf{C}}^{-1}_{(1,1)}\mathbf{x}_i + \mathbf{x}_i^{\mathrm{T}}\tilde{\mathbf{C}}^{-1}_{(1,2)}\boldsymbol{\Gamma}_i + \boldsymbol{\gamma}_i^{\mathrm{T}}\tilde{\mathbf{C}}^{-1}_{(2,1)}\mathbf{x}_i + \boldsymbol{\gamma}_i^{\mathrm{T}}\tilde{\mathbf{C}}^{-1}_{(2,2)}\boldsymbol{\Gamma}_i)^{-1}$
(the reason for keeping the˜notation here, is that in this case $\tilde{\Sigma}_{ii} \neq \Sigma_{ii}$.)
Furthermore

$$\begin{aligned} \mathbf{z} &= \boldsymbol{\Sigma}\boldsymbol{\tau} \\ \mu_i &= -\mathbf{z}^{\mathrm{T}}\tilde{\Sigma}_{ii}\beta\mathbf{X}^{\mathrm{T}}\mathbf{t} + \tilde{\Sigma}_{ii}\beta\mathbf{x}_i^{\mathrm{T}}\mathbf{t} \\ \boldsymbol{\mu} \rightarrow \boldsymbol{\mu}_{+i} &= \begin{bmatrix} \boldsymbol{\mu} - \mathbf{z}\mu_i \\ \mu_i \end{bmatrix} \end{aligned}$$

as well as

$$\begin{aligned} y_{m,1} &= \beta \mathbf{x}_m^{\mathrm{T}}\mathbf{x}_i + \lambda \boldsymbol{\gamma}_m^{\mathrm{T}}\boldsymbol{\Gamma}_i - (\beta \mathbf{x}_m^{\mathrm{T}}\mathbf{X} + \lambda \boldsymbol{\gamma}_m^{\mathrm{T}}\boldsymbol{\Gamma})\mathbf{z} \\ \tilde{\mathbf{S}}_m \rightarrow \tilde{\mathbf{S}}_{m,+i} &= \tilde{\mathbf{S}}_m - \tilde{\boldsymbol{\Sigma}}_{ii} y_{m,1}^2 \\ \tilde{\mathbf{Q}}_m \rightarrow \tilde{\mathbf{Q}}_{m,+i} &= \tilde{\mathbf{Q}}_m - \mu_i y_{m,1} \end{aligned}$$

– Update $\tilde{\boldsymbol{\Sigma}}$ when α_i changes

$$\boldsymbol{\Sigma} \rightarrow \boldsymbol{\Sigma}_{\text{new}} = \boldsymbol{\Sigma} - \boldsymbol{\Sigma}_i \kappa_i \boldsymbol{\Sigma}_i^{\text{T}} \tag{25}$$

where $\boldsymbol{\Sigma}_i = \boldsymbol{\Sigma}\mathbf{e}_i$ ($\mathbf{e}_i$ is the unit vector for the i-th dimension) and $\kappa_i = (\frac{1}{\alpha_{i,\text{new}} - \alpha_{i,\text{old}}} + \Sigma_{ii})^{-1}$.
Update $\boldsymbol{\mu} \rightarrow \boldsymbol{\mu}_{\text{new}} = \boldsymbol{\mu} - \boldsymbol{\Sigma}_i \kappa_i \mu_{i,2}$ in which $\mu_{i,2} = \boldsymbol{\Sigma}_i \mathbf{X}^{\text{T}} \beta \mathbf{t}$ and

$$\begin{aligned} y_{m,2} &= (\beta \mathbf{x}_m^{\text{T}} \mathbf{X} + \lambda \boldsymbol{\gamma}_m^{\text{T}} \boldsymbol{\Gamma}) \boldsymbol{\Sigma}_i \\ \tilde{\mathbf{S}}_m \rightarrow \tilde{\mathbf{S}}_{m,\text{new}} &= \tilde{\mathbf{S}}_m + \kappa y_{m,2}^2 \\ \tilde{\mathbf{Q}}_m \rightarrow \tilde{\mathbf{Q}}_{m,\text{new}} &= \tilde{\mathbf{Q}}_m + \kappa \mu_{i,2} y_{m,2} \end{aligned}$$

– Update $\tilde{\boldsymbol{\Sigma}}$ when α_i becomes infinity

$$\boldsymbol{\Sigma} \rightarrow \boldsymbol{\Sigma}_{-i} = \boldsymbol{\Sigma} - \frac{\boldsymbol{\Sigma}_i \boldsymbol{\Sigma}_i^{\text{T}}}{\Sigma_{ii}} \tag{26}$$

after which the i-th row and column need to be deleted. In the same spirit $\boldsymbol{\mu} \rightarrow \boldsymbol{\mu}_{-i} = \boldsymbol{\Sigma} \mathbf{X}^{\text{T}} \beta \mathbf{t}$ after which again the i-th row needs to be deleted and

$$\begin{aligned} y_{m,3} &= (\beta \mathbf{X}_m^{\text{T}} \mathbf{X} + \lambda \boldsymbol{\Gamma}_m^{\text{T}} \boldsymbol{\Gamma}) \boldsymbol{\Sigma}_i \\ \tilde{\mathbf{S}}_m \rightarrow \tilde{\mathbf{S}}_{m,-i} &= \tilde{\mathbf{S}}_m + \frac{y_{m,3}^2}{\Sigma_{ii}} \\ \tilde{\mathbf{Q}}_m \rightarrow \tilde{\mathbf{Q}}_{m,-i} &= \tilde{\mathbf{Q}}_m + \mu_i y_{m,3} \Sigma_{ii} \end{aligned}$$

6.5 Initialization

In [7] the initialization can be done by starting out with a single voxel in the model and then progressively adding voxels to the model. Another possibility is shown in [9] where the initial model has all voxels present with all $\boldsymbol{\alpha}$ set to a constant starting value of 1. In order to compare the performance of IRVoxM and RVoxM, we will in the following initialize as in [9].

7 Experiments

In order to demonstrate that our proposed optimizer outperforms RVoxM's, we will evaluate the performance of IRVoxM and RVoxM on a synthetic data. To make the comparison fair, we initialize the two algorithms identically with $\boldsymbol{\alpha} = \mathbf{1}$, $\beta = 1$ and $\lambda = 1$.

7.1 Experiments on synthetic data

We ran experiments on synthetic data. To model a single target value t, we generated a random vectorized image $\mathbf{x}$ by drawing random samples from a Gaussian distribution

Algorithm 1 IRVoxM algorithm

Initialize λ, β and all $\boldsymbol{\alpha}$ as in RVoxM [9].
loop
  loop
    Randomly pick a voxel i.
    Compute $\tilde{s}_i$,$\tilde{q}_i$ and a_i according to Eqs. 13 and 14.
    if $a_i \geq \tilde{s}_i$ **then**
      $\alpha_i = 0$
    else if $a_i < \tilde{s}_i$ **then**
      if $\tilde{s}_i - a_i < \tilde{q}_i^2$ **then**
        $\alpha_i = \frac{a_i(\tilde{s}_i + \tilde{q}_i^2) - \tilde{s}_i^2}{\tilde{s}_i - a_i - \tilde{q}_i^2}$
        if $\alpha_i < 0$ **then**
          $\alpha_i = 0.$
        end if
      else if $\tilde{s}_i - a_i \geq \tilde{q}_i^2$ **then**
        $\alpha_i = \infty$
      end if
    end if
    Update all quantities in an efficient manner as derived in 6.4.
  end loop
  Update β and λ by a simple search of the one-dimensional cost function.
end loop

with mean 0 and standard deviation 1 of size $M \times 1$. Using pre-determined constants $\boldsymbol{\alpha}_{\mathrm{true}} = (10^{12}\mathbf{v}, 0.5\mathbf{v}, 10^{12}\mathbf{v})^{\mathrm{T}}$, where $\mathbf{v}$ is a vector of ones and of dimension $\frac{M}{3} \times 1$, and $\lambda_{\mathrm{true}} = 10$, we constructed $\mathbf{P}_{\mathrm{true}} = \mathrm{diag}(\boldsymbol{\alpha}_{\mathrm{true}}) + \lambda_{\mathrm{true}}\boldsymbol{\Gamma}^{\mathrm{T}}\boldsymbol{\Gamma}$. Here, $\boldsymbol{\Gamma}$ is the incidence matrix for a 4-neighborhood. From $\mathbf{P}_{\mathrm{true}}$ we sampled weights $\mathbf{w}_{\mathrm{true}}$ and computed targets as $t = \mathbf{w}_{\mathrm{true}}^{\mathrm{T}}\mathbf{x} + \epsilon$, where the noise ϵ was sampled from a normal distribution with mean zero and inverse variance $\beta_{\mathrm{true}} = 10$. We constructed data this way for a varying number of training images N, yielding collections of image vectors $\mathbf{X}$ of size $N \times M$ as well as vectors of target values $\mathbf{t}$ of size $N \times 1$. We used an image size $M = 10 \times 10$. Lastly, we varied N from 10 to 100 and generated 100 independent pairs of $\mathbf{X}$ and $\mathbf{t}$ with the same weight vector $\mathbf{w}_{\mathrm{true}}$ for each value of N. For the test data, we generated another 100 independent pairs of $\mathbf{X}$ and $\mathbf{t}$ using $N = 100$, and applied the same weight vector $\mathbf{w}_{\mathrm{true}}$ as for the training data. Examples of two random images and the weight vector we used can be seen in Fig. 5. Fig. 6 shows the sparsity of the trained models, the training cost, which is the negative logarithm of the marginal likelihood given in Eq. 4, and the root mean square error (RMSE) between the true and the predicted target values computed on the test data sets. It also shows a comparison of the predicted and true weights by showing the l_2-norm of the difference between the true and the predicted weights of the two algorithms.

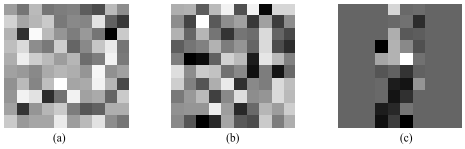

Fig. 5. Examples of two random images (a) and (b) as well as the weight vector (c) we used in our synthetic data experiment.

8 Discussion

The results reveal several weaknesses of the original RVoxM. First, while the true sparsity of our synthetic data is always 33% (since we set 1/3 of the 100 weights to be different from zero), RVoxM grossly overestimates the number of weights that are included in the model (see Figure 5 a). IRVoxM on the other hand produces sparser models, while still achieving a better training cost on the training data (see Figure 5 b). Hence IRVoxM is not over fitting to the training data, but finding sparse models that represent the data well. Furthermore, RVoxM and IRVoxM yield comparable RMSE on the test data with IRVoxM considerably outperforming RVoxM for larger N (see Figure 5 c). Finally, IRVoxM produces weights that are much closer to the true weights for all values of N (see Figure 5d). These results agree with our theoretical expectations and the experiments presented in [7].

9 Conclusion

We have re-visited the relevance voxel machine and introduced a better optimization scheme. By exploiting the form of the marginal likelihood function, we improved the way in which voxels are added and deleted from the sparse model during the optimization. Our algorithm IRVoxM outperforms RVoxM on synthetic data; it yields sparser models with good prediction performance and retains weight maps that are closer to the true synthetic weights than RVoxM's. Our aim was to show that our proposed algorithm IRVoxM improves over RVoxM's optimization scheme; thus we compared the two algorithms side by side. Our new optimization strategy performs as anticipated, and opens up a whole new avenue for speeding up computations, as was done previously for RVM [4] by FastRVM [7]. One key problem of RVoxM is the computational burden, especially during the first few iterations, where computational time is cubic in the number of voxels. IRVoxM does not need to be initialized with all voxels (as has been done for comparison to RVoxM in all our experiments here). One can start with only a few voxels in the model, which reduces the computational cost tremendously and preliminary experiments show that this approach performs equally well. Furthermore, our explicit functional formulation of the marginal likelihood function for a single α_i makes it possible to sample from the hyperparameters distributions, which had not been possible with RVoxM.
In further versions of IRVoxM, we plan to implement a different initialization strategy

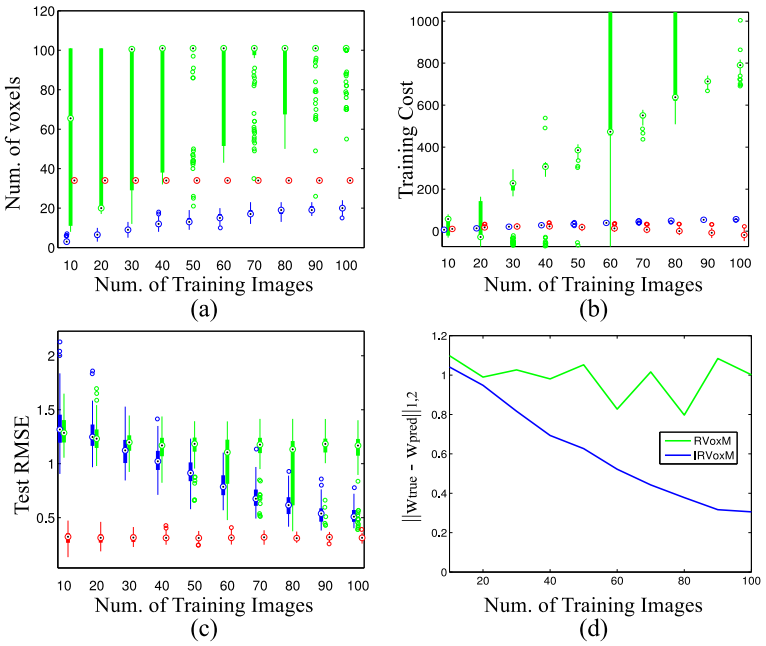

Fig. 6. Results for the synthetic data showing the resulting training sparsity (a), the training cost (b) and the root mean square error (RMSE) on the test data (c) for 100 independent repetitions. The box plots in (a), (b) and (c) show the ground truth (red), RVoxM (green) and IRVoxM (blue). Filled black dots indicate the median, filled boxes extend to the most extreme values within 1.5 times the interquartile range of the box. Lines extend to the adjacent value. Samples beyond those points are marked with colored circles. In (d) we show the l_2-norm of the differences between the true weights and the weights RVoxM produces (green) and the true weights and the weights IRVoxM produces (blue).

that enables us to increase the speed of IRVoxM, as well as exploit the possibility of sampling from the hyperparameter distribution.

10 Acknowledgments

This research was supported by the Alfred Benzon and the Lundbeck Foundation and carried out in whole or in part at the Athinoula A. Martinos Center for Biomedical Imaging at the Massachusetts General Hospital, using resources provided by the Center for Functional Neuroimaging Technologies, P41EB015896, a P41 Biotechnology Resource Grant supported by the National Institute of Biomedical Imaging and Bioengineering (NIBIB), National Institutes of Health. This work also involved the use of instrumentation supported by the NIH Shared Instrumentation Grant Program and/or High-End Instrumentation Grant Program; specifically, grant number(s) S10RR023401, S10RR019307, S10RR019254 and S10RR023043.

References

1. Mackay, D.J.C.: Bayesian methods for adaptive models. PhD thesis, Pasadena, CA, USA (1992)
2. Tipping, M.: The relevance vector machine. In: Advances in Neural Information Processing Systems 12, MIT Press (2000) 652 – 658
3. Bishop, C.M., Tipping, M.E.: Variational relevance vector machines. In: Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence. UAI '00, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (2000) 46–53
4. Tipping, M.: Sparse bayesian learning and the relevance vector machine. The Journal of Machine Learning Research **1** (2001) 211–244
5. Rasmussen, C.E.: Healing the relevance vector machine through augmentation. In: In Proc. of the 22nd International Conference on Machine learning (ICML 2005), ACM Press (2005) 689–696
6. Schmolck, A., Everson, R.: Smooth relevance vector machine: a smoothness prior extension of the rvm. Machine Learning **68**(2) (2007) 107–135
7. Tipping, M.E., Faul, A.: Fast marginal likelihood maximisation for sparse bayesian models. In: Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics. (2003) 3–6
8. Sabuncu, M., Leemput, K.: The relevance voxel machine (rvoxm): A bayesian method for image-based prediction. In Fichtinger, G., Martel, A., Peters, T., eds.: Medical Image Computing and Computer-Assisted Intervention MICCAI 2011. Volume 6893 of Lecture Notes in Computer Science., Springer Berlin Heidelberg (2011) 99–106
9. Sabuncu, M., Van Leemput, K.: The relevance voxel machine (rvoxm): A self-tuning bayesian model for informative image-based prediction. Medical Imaging, IEEE Transactions on **31**(12) (2012) 2290–2306